CENTRE DE RECHERCHE SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A.
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: 05-24146 BIBPOLYTEC
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

90-06418/

CRT - 702

A MODEL FOR THE DYNAMIC TRAFFIC ASSIGNMENT PROBLEM

by

Omar Drissi-Kaïtouni

This research was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

Université de Montréal
Centre de recherche sur les transports - Publication #702
June 1990

Dépôt légal - 2e trimestre 1990

ABSTRACT

This paper describes a Dynamic Traffic Assignment Model, aimed at simulating the flows of urban transportation networks, mainly during the peak periods. The approach that we propose, is mainly based on the assumption that the users know at each period of time, the travel times on the links of the network, but not for the next periods, and that, at each node, they enter their actual shortest path; in other words, at the beginning of each period of time, a user is assigned on the first link of the actual shortest path to his destination. Numerical tests have been performed with the transportation network of the city of Hull (Quebec, Canada), and the results are encouraging.

Key words: Dynamic, Traffic, Network, Equilibrium, Assignment.

RÉSUMÉ

Cet article décrit un modèle d'affectation dynamique du trafic, pour la simulation des flux véhiculaires, sur un réseau de transport urbain, durant les heures de pointes. L'approche que nous proposons, repose principalement sur l'hypothèse que les usagers connaissent à chaque instant les temps actuels de voyage sur les liens du réseau, et s'engagent sur les chemins les plus courts les conduisant à leurs destinations; ainsi, à chaque noeud du réseau, les usagers ont la possibilité de modifier leurs chemins. Les résultats que nous avons obtenus avec le réseau de la ville de Hull (Québec, Canada) sont encourageants.

Mots clés: Dynamique, trafic, réseau, équilibre, affectation.

1. INTRODUCTION

This paper is concerned with the models aimed at simulating the flows of urban transportation networks, mainly during the peak periods. In the literature, these models are referred to as the Traffic Assignment Models (TAM); they all fit in one of two main classes that are the Static and the Dynamic Traffic Assignment Models (respectively STAM and DTAM).

1.1 The Static Traffic Assignment Models

During the past forty years, most of the research has been concentrated in this class of models where it is assumed that the demand is constant over the time; a global demand for a given horizon of time (a few hours, a day, ...) is assigned on the network in order to minimize the individual transportation costs (user optimal assignment). Among the most relevant STAM's proposed in the literarture, we must mention the models of Wardrop (1952) for uncongested networks, of Dafermos and Sparrow (1969) for congested networks, and of Guélat et al. (1987) for multimode - multiproduct congested networks (yet, this last one has been designed for the analysis of intercity freight flows).

Altough the demand is not constant over the time in an urban context, the STAM's have been extensively used in practice; in fact, even if these models are not suitable to analyze the flow variations over the study horizon of time, when well calibrated, they give a very good approximation of the total flow on the links during the whole study horizon of time.

1.2 The Dynamic Traffic Assignment Models

These models are more suitable than the static one, to analyze the flow variations on the links during the peak periods since the demand is allowed to vary in time. Unfortunately, most of the models of this class, proposed in the literature, incorporate important limitations. In the following, we give some brief comments on the most relevant DTAM's.

Some models are too restrictive such as that of Robillard (1974) that does not apply to capacity constrained links (e.g. congested networks). Others (Hendrickson and Kocur (1981), Arnott et al. (1987) and de Palma et al. (1987)) consider only parallel paths networks where congestion occurs at a single link (bottleneck).

Among less restrictive models, there is that of Yagar (1970) where, at each period, the demand between an O/D pair is assigned on the actual shortest path connecting that pair; when the demand is larger than the capacity of any link of the path, the excess is added to the demand of the next period. The same approach is used by Leonard et al. (1978) in the computer system Contram. The main objection to this approach is that the demand is assigned on paths, which means that the user decides

at the origin node, on the route to be followed to reach the destination; this is not in accordance with the assumption that *at any period, the user knows the state of the network*, which implies that the user can modify the route, for a better one, while progressing to the destination.

Since 1978, many optimization models have been proposed in the literature for the DTAM. In the following, we review briefly their underlying concepts.

In 1978, Merchant and Nemhausser present a first minimization model for the single origin-destination DTAP. Ho (1980) proposes a stepwise linear version of this model which has been more deeply investigated by Carey (1986 and 1987); this author presents a convex version of the model and some techniques to generalize it to the many origins and many destinations case. Finally, Friesz et al. (1988) propose a continuous time version (optimal control problem). In these models, the authors represent the links' dynamic of flow by:

$$x_a^{t+1} = x_a^t + d_a^t - g_a(x_a^t) \;\; \forall\, a \in A, t = 0, 1, .., T-1$$

where:

x_a^t and x_a^{t+1} are the flows on link a at periods t and $t+1$,

d_a^t is the flow that enters link a at period t,

$g_a(x_a^t)$ is the flow that exits link a at period t,

A is the set of links of the network, and

T is the number of periods of study (horizon).

Apart from the fact the model leads to huge mathematical programs, its authors impose too strong conditions on the exit and cost functions:

- the exit functions $g_a()$ are restricted to be concave in order to obtain convex programs. Such a condition does not necessary hold in congested networks.

- the dynamic flow conservation constraints of the model do not ensure that the flows will reach the destination (independently of the horizon T); in order to overcome this problem, the authors of the model impose an unrealistic condition on the link cost functions, that is roughly that the costs of the links must increase with their distance to the destination.

A few years ago, Zawack and Thompson (1987) adapted to the DTAM the model proposed for the Dynamic Maximal Flow Problem by Ford and Fulkerson (1962) and Fulkerson (1975). In this model, each link $a \in A$ has a capacity and a fixed traversal time. For any period of time, at each node vehicles can either be transshipped on exiting links or wait until the next period. The resulting model is a transshippment problem over an expanded (space-time) network where only one destination is allowed.

Among the lacuna of this model, the queues hold at the nodes rather than on the links, which means that the intersections of the city are viewed as stacks for the vehicles.

Recently, Drissi and Hameda (1989) formulated this model in terms of path flows (in the space-time expanded network) as a multicommodity network equilibrium prob-

lem. Further, they modelized the expanded space-time network in a way to include the queues on the links rather than on the nodes.

Anyway, all the optimization models that have been proposed in the literature for the DTAM have the characteristic of leading to huge mathematical programs, hence can difficultly be used in practice to analyze real city networks. On the other hand, some assumptions underlying these models, such as that the user knows *a priori* the state of the network (volumes and travel times on all the links) for all the periods of the study horizon, may be debatable in the context of continually evolving networks.

In the following we present a Dynamic Traffic Assignment Model that combines, under realistic assumptions, the good properties of the heuristic models (e.g. require to solve only relatively small subproblems at each iteration) and of the optimization models (rational choice of the routes by the users). In section 2, we describe our model for one destination, and, in section 3, we adapt the results for the multicommodity case. In section 4, we present numerical results for the city of Hull (Quebec) in Canada, and we conclude the paper in section 5.

2. THE MODEL

Let's define first our notation:

2.1 Notation

q	:	the unique destination node considered in this section.
v_a^t	:	flow on link a at the beginning of period t.
v^t	:	vector of the link flows at the beginning of period t.
u_a^t	:	flow that exits link a during period t.
u^t	:	vector of the exit flows at the beginning of period t.
d_a^t	:	flow that enters link a during period t.
d^t	:	vector of the entering flows at the beginning of period t.
e_l^t	:	new demand at node l during period t.
e_l	:	vector of the new demand during period t.
$c_a(v_a^t)$	:	travel time function[1] on link a.
s_l^t	:	cost of the shortest path[2] from node l to q at the beginning of t.
A	:	set of the links of the network.
N	:	set of the nodes of the network.
A_l	:	set of the links having l as tail node.
B_l	:	set of the links having l as head node.

[1] These functions may also be generalized transportation cost functions.

[2] The shortest path is computed according to the travel times on the links at the beginning of the current period t.

2.2 The Dynamic Traffic Assignment Model

Our model is based on the following assumptions:

A1 *At any time period, the user knows the state of the network, e.g. knows the travel times on all the links of the network for the current period, but not for the next periods.*

A2 *At any time period, each user strives to minimize his own travel time.*

In order to have a match between the moment when the users re-evaluate their routes (the end of each period) and the moment when they can effectively change them (when they are at a node), we assume that:

A3 *All the links have (say approximatively) the same free traversal time (e.g. traversal time when no congestion is present) that is equal to one period of time.*

This assumption is not very restrictive; the long links may be "broken" into smaller links of appropriate length, and the too short links may be aggregated to others to form again links of appropriate length. Finally:

A4 *For each link a, there exists an exit function $g_a(x)$ that defines the exiting flow.*

We do not impose any restriction on the shape (concavity, ...) of the functions $g_a(x)$, neither on the variable x that can simply be the current flow v_a^t on link a, or the vector of current flows v^t on all the network, or a more complicated variable. These functions may be calibrated *in situ*, or constructed from the travel time functions as shown in the Appendix 1. However, we assume that:

A5 *The flow that enters a link at a given period must exit the link after a finite number of periods.*

A corollary to assumption A3, is that the travel time functions $c_a(v_a^t)$ satisfy: $c_a(0) = 1$ period of time, and $c_a(v_a^t) \geq 1$ for $v_a^t \geq 0$.

Now, let's define, for each link $a = (i, j)$, at any time period t, the travel cost:

$$C_a^t = c_a(v_a^t) + s_j^t,$$

that is the estimation, at the beginning of period t, of the minimum travel time from the node i (tail node of link a) to the destintion q, when the link a is used. According to our assumptions, at each time period, the users that exit a link $a = (i, j)$, will seek to enter the link $b = (j, k)$ in A_j that minimizes C_b^t.

Then, the mathematical model corresponding to each period of time t may be stated as follows:

$$(P_1^t)\begin{cases} \min \sum_{a\in A}\int_0^{v_a^{t+1}} C_a^t\,dx \\ s.t. \quad \sum_{a\in B_l} u_a^t + e_l^t = \sum_{a\in A_l} d_a^t \quad \forall\, l\in N \\ d_a^t \geq 0 \quad \forall\, a\in A \end{cases}$$

$$\text{with the implicit constraints:} \begin{cases} v_a^{t+1} = v_a^t + d_a^t - u_a^t \quad \forall\, a\in A \\ u_a^t \text{ is given explicitly by the function } g_a() \end{cases}$$

We must note here, that the variables of the program (P_1^t) are $d_a^t, a \in A$. $v_a^t, a \in A$ are the flows (known) at the beginning of period t, $u_a^t, a \in A$ are the exiting flows during period t (given explicitly by the exit functions), and $v_a^{t+1}, a \in A$ are given explicitly by the first implicit constraint.

(P_1^t) is a linear program, hence, its optimal solution is an extremal solution, and it is easy to see that it is obtained by:

$$\forall\, a=(i,j)\in A,\quad d_a^t = \begin{cases} \sum_{b\in B_i} u_b^t + e_i^t & \text{if } C_a^t = \min_{b\in A_i}\{C_b^t\} \\ 0 & \text{otherwise} \end{cases}$$

If we define w_i^t to be $\sum_{b\in B_i} u_b^t + e_i^t \quad \forall\, i \in N$, this relation can be rewritten:

$$\forall\, a=(i,j)\in A,\quad d_a^t = \begin{cases} w_i^t & \text{if } C_a^t = \min_{b\in A_i}\{C_b^t\} \\ 0 & \text{otherwise} \end{cases}$$

The link $a = (i,j)$ that satisfies $C_a^t = \min_{b\in A_i}\{C_b^t\}$ is simply the first link of the shortest path from the node i to the destination q.

The Dynamic Traffic Assignment algorithm corresponding to this model can then be stated simply as: at each time period t, the flows u_a^t, exiting from the links $a \in B_j$, and the new demand e_j^t at node j, are transshipped to the link (j,k) that belongs to the (current) shortest path from the node j to the destination q. Let's define a node $\bar{n} = |N|+1$, and a fictive link (a sink) $\bar{a} = (q, \bar{n})$. The algorithm then takes the form:

Algorithm A1-1

Step 0- Initialize v^0, and set $t = 0$.

Step 1- WHILE (Stopping Criterion not satisfied) DO: (process period t)

i- Compute the actual travel times $c_a(v_a^t)$ $\forall\, a \in A$.

ii- Find the tree $T(q,t)$ of shortest paths from all the nodes to destination q, and add the fictive link $\bar{a}$ to $T(q,t)$.

iii- Compute u^t by using the exit functions.

iv- Initialize $v^{t+1} = v^t$.

v- FOR each node $j \in N$ DO:

α- Let $b = (j,k)$ the link in $T(q,t)$ exiting from j.

β- Let $v_b^{t+1} = v_b^{t+1} + \sum_{a \in B_j} u_a^t + e_j^t$.

γ- Let $v_a^{t+1} = v_a^{t+1} - u_a^t$ $\forall\, a \in B_j$.

vi- $t = t + 1$.

In the following, we would like to comment each of the steps of the algorithm, and then, we give a proof of convergence.

-Step 0- The flows on the links are initialized to their value at the first period of study; these may be null or not.

-Step 1-: This is the general step which is repeated for each period of time. We suggest to stop the algorithm when the total demand on the network is smaller than a certain given value (that is $\|v^t\| \leq$ MinValue), or simply when the number of processed period attains a fixed study horizon T.

-Step 1-i-: In this step, we compute the actual link travel times needed to find the shortest paths. The link travel times are evaluated by using calibrated link travel time functions (see for instance, in Appendix 1, the classes of functions used for the city of Hull, Canada).

-Step 1-ii-: According to assumption A1, the tree of shortest paths is defined using the link travel times at the current period of time (computed in step 1-i).

-Step 1-iii-: The flow exiting a link is computed, according to assumption A4, by using exit functions (see for instance Appendix 1).

-Steps 1-iv-v-: The transfer of flow is performed at each node: the arriving flow to a node, is transferred to the first link of the current shortest path between the given node and the destination.

Theorem 1: *If the demand is $e^t = 0$ for all $t \geq \bar{t}$, where $\bar{t}$ is a finite time period, then all the flows generated by the algorithm A1-1 reach the destination q in a finite number of periods.*

Proof: $T(q,t)$ being the tree of shortest paths from all the nodes to q at the period t, let's define DT^t to be the total flow on the network at the period t.

In the following, we consider an arbitrary time period t that is greater than or equal to $\bar{t}$, and we suppose that $DT^t > 0$ (otherwise there would be nothing to prove; all the flows would have reached the destination). Then, we show that in a finite number of periods, DT^t falls to zero.

Let n^t be the lowest level in $\mathrm{T}(q,t)$ (e.g. n^t the closest to 0) such that there exists a node $i \in N$ (at this level n^t), that has a positive input flow $w_i^t > 0$. Two cases arise:

-Case 1: $n^t = 0$: this is the level of the root q of the tree $\mathrm{T}(q,t)$, so, $w_q^t > 0$ is dropped of the network at this period; then $DT^{t+1} = DT^t - w_q^t < DT^t$. If $DT^{t+1} = 0$, the proof is complete, all the flows have reached the destination. Otherwise, the total flow on the network has been decreased during the period t; in this case we set $t = t + 1$, and we restart.

-Case 2: $n^t > 0$: Let i be any node at the level n^t, such that $w_i^t > 0$. At the beginning of the period $t+1$, w_i^t is transferred to the link $a = (i,j)$ in the tree $\mathrm{T}(q,t)$; then $w_j^{t+1} > 0$, and the level of the node j is $n^t - 1$ in $\mathrm{T}(q,t)$ during period t. In the following we show that the level of the node j is $n^t - 1$ in $\mathrm{T}(q,t+1)$ too. Therefore, $n^{t+1} = n^t - 1$, and in a finite number k of periods, $n^{t+k} = 0$. This means that in a finite number of periods, it is possible to draw a part of w_i^t to the destination q. Furthermore, by using assumption A5, and the fact that the size of the network is finite, then the whole quantity w_i^t is drawn to the destination q in a finite number of periods.

Now, we must show that the level of the node j is $n^t - 1$ in $\mathrm{T}(q,t+1)$ too. In order to prove this, we show that the trees $\mathrm{Tn}(q,t)$ and $\mathrm{Tn}(q,t+1)$ (that are the trees $\mathrm{T}(q,t)$ and $\mathrm{T}(q,t+1)$ restricted to the levels 0, 1, ..., $n^t - 1$) are identical.

During the period t, $w_l^t = 0$ for all the nodes l with a level less than or equal to $n^t - 1$, in $\mathrm{T}(q,t)$. Then, $\forall\, a \in \mathrm{Tn}(q,t)$, $v_a^t = 0$ and $v_a^{t+1} = 0$, and, $c_a(v_a^t) = 1$ and $c_a(v_a^{t+1}) = 1$ (we recall here that the value of 1 is the minimal value of a link travel time). Then, during the period t, s_l^t = level of l in $\mathrm{Tn}(q,t) \leq n^t$, for any node l in $\mathrm{Tn}(q,t)$.

Let's show the equality of the two (sub)trees $\mathrm{Tn}(q,t)$ and $\mathrm{Tn}(q,t+1)$ by recursivity:

-1- n=0: Let $N_n^t = \{l \in N :$ level of l in $\mathrm{Tn}(q,t) = n\}$. Clearly $N_0^t = \{q\}$ and $N_0^{t+1} = \{q\}$. So $\mathrm{Tn}(q,t)$ and $\mathrm{Tn}(q,t+1)$ are identical at the level 0. We have also that $s_q^t = s_q^{t+1} = 0$.

-2- $0 < n < n^t$: By recursivity, $N_k^t = N_k^{t+1}$ for all the levels $k < n$. We have also, by recursivity, that $\forall\, l \in N_k^t$, $s_l^t = s_l^{t+1} = k$, $\forall\, k < n$.

Now let's consider any node $l \in N_n^t$. Since l is not in any of the sets $N_k^t = N_k^{t+1}$, $k < n$, then $s_l^{t+1} \geq n$. But the flow volumes, and then, the travel times on all the links of the tree $\mathrm{Tn}(q,t)$ remaining the same during the periods t and $t+1$, the shortest path from l to q is in the period $t+1$, the same as in the period t, with $s_l^{t+1} = s_l^t = n$. It follows from this that $N_n^t \subseteq N_n^{t+1}$.

In order to prove that $N_n^{t+1} = N_n^t$, it remains to show that $N_n^{t+1} \subseteq N_n^t$. Let's then consider any node $l \in N_n^{t+1}$. Let $a = (l,l')$ be the first link of the shortest path from l to q in $\mathrm{Tn}(q,t+1)$. Then $l' \in N_{n-1}^{t+1}$ and $s_{l'}^{t+1} = n-1$. But $N_{n-1}^{t+1} = N_{n-1}^t$, so l' is in

$\mathrm{Tn}(q,t)$, and then, $v_a^t = 0$ and $c_a(v_a^t) = 1$. It follows then that $s_l^t = n - 1 + 1 = n$, which shows that $l \in N_n^t$. Consequently, $N_n^{t+1} \subseteq N_n^t$ and since $N_n^{t+1} \supseteq N_n^t$, it follows that $N_n^{t+1} = N_n^t$.□

2.3 Refinement of the model

In the model described above, it has been noticed that the optimal solution of each subproblem (P_1^t) is extremal, that is, at each node j, the input flow is transferred in whole to the link (j,k) that belongs to the current shortest path from j to the destination. This transfer can eventually increase the travel time on the link (j,k) so some paths using other links (j,l) in A_j, may become competitive. The refinement that we propose here is to assign the input flow on selected links $b = (j,l)$ in A_j, so that the travel time on the paths that use these links be equal and minimum among all the paths between j and the destination. The corresponding mathematical program, for a period t, is the following:

$$(P_1'^t)\begin{cases} \min \quad \displaystyle\sum_{a\in A}\int_0^{v_a^{t+1}} C_a^t(x)\,dx \\ s.t. \quad \displaystyle\sum_{a\in B_l} u_a^t + e_l^t = \sum_{a\in A_l} d_a^t \quad \forall\, l \in N \\ \qquad d_a^t \geq 0 \quad \forall\, a \in A \end{cases}$$

with the implicit constraints: $\begin{cases} v_a^{t+1} = v_a^t + d_a^t - u_a^t \;\forall\, a \in A \\ u_a^t \text{ is given explicitly by the function } g_a() \end{cases}$

where: $C_a^t(x) = c_a(x) + s_j^t \;\; \forall\, a = (i,j) \in A$.

w_l^t being the input flow $\displaystyle\sum_{a\in B_l} u_a^t + e_l^t$ for each node l during the period t, if we define λ^t to be the vector of the dual variables corresponding to the equality constraints $w_l^t = \displaystyle\sum_{a\in A_l} d_a^t \;\; \forall\, l \in N$ in $(P_1'^t)$, then the optimality conditions are:

$$\begin{cases} C_a^t(v_a^{t+1}) - \lambda_i^t \geq 0 & \forall\, a = (i,j) \in A, \\ (C_a^t(v_a^{t+1}) - \lambda_i^t)d_a^t = 0 & \forall\, a = (i,j) \in A, \\ d_a^t \geq 0 & \forall\, a = (i,j) \in A. \\ w_l^t = \displaystyle\sum_{a\in A_l} d_a^t & \forall\, l \in N, \end{cases}$$

and the implicit constraints as above.

These conditions may be rewritten:

$$\begin{cases} \forall\, a=(i,j)\in A \quad C_a^t(v_a^{t+1}) \begin{cases} = \lambda_i^t & \text{if } d_a^t > 0 \\ \geq \lambda_i^t & \text{if } d_a^t = 0 \end{cases} \\ \forall\, l \in N \qquad\qquad w_l^t = \sum_{a\in A_l} d_a^t, \end{cases}$$

and the implicit constraints as above.

Finally, since each link has exactly one tail node, these conditions can be decomposed by nodes:

$$\forall\, l \in N \begin{cases} \forall\, a \in A_l \;\; C_a^t(v_a^{t+1}) \begin{cases} = \lambda_l^t & \text{if } d_a^t > 0 \\ \geq \lambda_l^t & \text{if } d_a^t = 0 \end{cases} \\ w_l^t = \sum_{a\in A_l} d_a^t, \end{cases}$$

and the implicit constraints as above.

According to these conditions, we meet our expectation, that is that at each node l, the input flow w_l^t is distributed on selected links $a \in A_l$ so that the travel time (evaluated at this moment) on the paths that use these links be equal and minimum among all the paths that join l and the destination q.

The corresponding algorithm may be stated as follows:

Algorithm A1-2

Step 0- Initialize v^0, and set $t = 0$.

Step 1- WHILE (Stopping Criterion not satisfied) DO: (process period t)

i- Compute the actual travel times $c_a(v_a^t)$ $\forall\, a \in A$.

ii- Find the tree $T(q,t)$ of shortest paths from all the nodes to destination q, and add to $T(q,t)$, the fictive link $\bar{a}$.

iii- Compute u^t by using the exit functions.

iv- FOR each node $l \in N$ DO:

α- Let the input flow to l be $w_l^t = \sum_{a\in B_l} u_a^t + e_l^t$.

β- FOR each $a \in A_l$, compute d_a^t and v_a^{t+1}, such that:

- $\sum_{a\in A_l} d_a^t = w_l^t$

- $C_a^t(v_a^{t+1}) \begin{cases} = \lambda_l^t & \text{if } d_a^t > 0 \\ \geq \lambda_l^t & \text{if } d_a^t = 0 \end{cases}$

  where $\lambda_l^t = \min_{a \in A_l}\{C_a^t(v_a^{t+1})\}$

  and $v_a^{t+1} = v_a^t + d_a^t - u_a^t \;\; \forall a \in A_l$

v- $t = t + 1$.

As in the preceding subsection, we may prove that this algorithm converges:

Theorem 2: *If the demand is $e^t = 0$ for all $t \geq \bar{t}$, where $\bar{t}$ is a finite time period, then all the flows generated by the algorithm A1-2 reach the destination q in a finite number of periods.*

Proof: The arguments of the proof are identical to those used to prove Theorem 1; in effect, in the algorithm A1-1, the input flow w_i^t, that is at the level n^t in $T(q,t)$, is transferred to the link $a = (i,j)$ that belongs to $T(q,t)$; then we showed that $n^t > n^{t+1} = n^t - 1 > ...$, so, after a finite number of periods, a part of the flow exits of the network. In A1-2, w_i^t is (eventually) transferred to many links (i,j_i) in A_i. The same arguments hold for the algorithm A1-2, if we replace the node j (in $a = (i,j)$) by the node j' where:

$$j' \equiv \min_{i \in N}\{\min_{j_i \in A_i}\{level(j_i)\}\}$$

In this case, $n^{t+1} \leq n^t - 1$ instead of $n^{t+1} = n^t - 1$, and the remainder of the proof is based on the same arguments as in the proof of Theorem 1. □

3. THE MULTI-DESTINATION CASE

In this section, we adapt the results of section 2 to the case where many destinations are present in the network. The notation that we use is the same as in section 2, but new variables must be introduced in order to take into account the multiplicity of the destinations:

3.1 New variables

D	:	the set of destination nodes.
$v_{a,q}^t$	:	portion of v_a^t going to the destination q.
$u_{a,q}^t$	:	portion of u_a^t going to the destination q.
$d_{a,q}^t$	:	portion of d_a^t going to the destination q.
$e_{l,q}^t$	:	new demand at node l for destination q, during period t.
$s_{l,q}^t$	:	cost of the shortest path from node l to q at the beginning of period t.

3.2 The model

In the previous section, we assumed that the exiting flow of a link a is given by means of an exit function $g_a()$. But, as complex as $g_a()$ can be, it is not reasonable

to assume that it can give the exit flow by destination; an exit function usually gives the total flow that exits of a link, but says nothing on the destinations of that flow. In the following, we assume that the portion $u^t_{a,q}$ of the exiting flow u^t_a that goes to the destination q is proportional to the portion of the flow on the link going to the same destination:

$$p^t_{a,q} = \frac{v^t_{a,q}}{\sum_{q' \in D} v^t_{a,q'}} = \frac{v^t_{a,q}}{v^t_a}.$$

Then,

$$u^t_{a,q} = p^t_{a,q} u^t_a = \frac{v^t_{a,q}}{v^t_a} g_a().$$

Under this basis, the generalization of the model P^t_1 to the multi-destination case can be stated as:

$$(P^t_D) \begin{cases} \min \ \sum_{a \in A} \sum_{q \in D} \int_0^{v^{t+1}_{a,q}} C^t_{a,q}\, dx \\ s.t. \ \sum_{a \in B_l} u^t_{a,q} + e^t_{l,q} = \sum_{a \in A_l} d^t_{a,q} \ \ \forall\, l \in N, \ \ \forall\, q \in D \\ d^t_{a,q} \geq 0 \ \ \forall\, a \in A, \ \ \forall\, q \in D \end{cases}$$

with the implicit constraints:
$$\begin{cases} v^{t+1}_{a,q} = v^t_{a,q} + d^t_{a,q} - u^t_{a,q} \ \ \forall\, a \in A, \ \ \forall\, q \in D \\ u^t_{a,q} = p^t_{a,q} u^t_a = \frac{v^t_{a,q}}{v^t_a} g_a() \end{cases}$$

The functions (constants) appearing in the objective function are defined, for each link $a = (i, j)$, as in the 1-destination case as:

$$C^t_{a,q} = c_a(v^t_a) + s^t_{j,q}.$$

The problem is clearly separable by destinations, which leads to the following adaptation of the algorithm A1-1:

Algorithm AD-1

Step 0- Initialize v^0, and set $t = 0$.

Step 1- WHILE (Stopping Criterion not satisfied) DO: (process period t)

a- Compute the actual travel times $c_a(v^t_a) \ \ \forall\, a \in A$.

b- FOR each destination $q \in D$, DO:

i- Find the tree $T(q,t)$ of shortest paths from all the nodes to destination q, and add the fictive link $\bar{a} = (q, \bar{n})$ to $T(q,t)$.

ii- Compute $u^t_{\bar{a}}$ by using the exit functions and the relations above.

iii- Initialize $v^{t+1}_q = v^t_q$.

iv- FOR each node $j \in N$ DO:

α- Let $b = (j,k)$ the link in $T(q,t)$ exiting from j.

β- Let $v^{t+1}_{b,q} = v^{t+1}_{b,q} + \sum_{a \in B_j} u^t_{a,q} + e^t_{j,q}$.

γ- Let $v^{t+1}_{a,q} = v^{t+1}_{a,q} - u^t_{a,q} \ \forall a \in B_j$.

c- $t = t + 1$.

In this algorithm, each portion of the flow exiting a link, that is going to a given destination q is assigned in whole to the first link of the shortest path to the destination q, at the current period of time. The version where the exiting flow is equilibrated between the exiting links may be stated as follows:

$$(P'_D) \begin{cases} \min \ \sum_{a \in A} \sum_{q \in D} \int_0^{d^t_{a,q}} C^t_{a,q}(v^t_a - u^t_{a,q} + x)dx \\ s.t. \ \sum_{a \in B_l} u^t_{a,q} + e^t_{l,q} = \sum_{a \in A_l} d^t_{a,q} \ \forall l \in N, \ \forall q \in D \\ d^t_{a,q} \geq 0 \ \forall a \in A, \ \forall q \in D \end{cases}$$

with the implicit constraints:
$$\begin{cases} v^{t+1}_{a,q} = v^t_{a,q} + d^t_{a,q} - u^t_{a,q} \ \forall a \in A, \ \forall q \in D \\ u^t_{a,q} = p^t_{a,q} u^t_a = \frac{v^t_{a,q}}{v^t_a} g_a() \end{cases}$$

where the functions appearing in the objective function are defined, for each link $a = (i,j)$, as:

$$C^t_{a,q}(y) = c_a(y) + s^t_{j,q}.$$

The corresponding algorithm, that is a generalization of the algorithm A1-2, can be stated as:

Algorithm AD-2

Step 0- Initialize v^0, and set $t = 0$.

Step 1- WHILE (Stopping Criterion not satisfied) DO: (process period t)

a- Compute the actual travel times $c_a(v^t_a) \ \forall a \in A$.

b- FOR each destination $q \in D$, DO:

i- Find the tree $T(q,t)$ of shortest paths from all the nodes to destination q, and add the fictive link $\bar{a} = (q, \bar{n})$ to $T(q,t)$.

ii- Compute $u^t_{\bar{a},q}$ by using the exit functions and the relations above.

iii- FOR each node $j \in N$ DO:

α- Let the input flow to j for q, be $w^t_{j,q} = \sum_{a \in B_j} u^t_{a,q} + e^t_{j,q}$.

β- FOR each $a \in A_j$, compute $d^t_{a,q}$ and $v^{t+1}_{a,q}$, such that:

- $\sum_{a \in A_j} d^t_{a,q} = w^t_{j,q}$

- $C^t_a(v^t_a - u^t_{a,q} + d^t_{a,q}) \begin{cases} = \lambda^t_j & \text{if } d^t_{a,q} > 0 \\ \geq \lambda^t_j & \text{if } d^t_{a,q} = 0 \end{cases}$

where $\lambda^t_j = \min_{a \in A_j} \{C^t_a(v^t_a - u^t_{a,q} + d^t_{a,q})\}$

and $v^{t+1}_{a,q} = v^t_{a,q} + d^t_{a,q} - u^t_{a,q} \quad \forall\, a \in A_j$

iv- $t = t + 1$.

It is not possible to state a theorem similar to Theorems 1 and 2 for the multi-destination case, since the arguments used in their proofs (that the tree of shortest paths does not change for some levels, which allows the flow to step forward to the destination) do not hold in this case. Ultimately, it is possible to construct pathological cases where the flows cycle indefinitely, but the cycling conditions (on the topology of the network, the flow volumes and on the synchronization of the different commodities moves on the cycles) are unlikely to occur in real networks. Further more, it is possible to include in the computer program, a strategy that prevents cycling by overcoming the synchronization condition (for instance, by modifying, from time to time, the order in which the commodities are scanned). As for our numerical results, all the flows reach all the destinations without using such a strategy.

4. NUMERICAL RESULTS

In order to test the validity of our model, we have used available data for the city of Hull (Quebec, Canada) at the Centre de recherche sur les transports, Université de Montréal.

The network has 798 links, 501 nodes and 142 O/D pairs. The total demand between the origins and the destinations is given in Appendix 2, and the classes of travel time functions in Appendix 1.

Unfortunately, we do not have the flows on the links for each period of time, but only the total flow on the links. Therefore, in our numerical tests, we compared this value with the total flow on the links (the sum of the flow on the links of each period) obtained by the Dynamic Traffic Assignment Model described in section 3 (see Figure 1). The regression line between the two sets of flows is:

Dynamic Total Flow = 1.03 * Total Flow - 8.4

and when the constant term is constrained to be zero, the equation is:

Dynamic Total Flow = 1.02 * Total Flow

We note that it is reasonable to assume that the constant term is zero since the mean flow is about 170, and the standard deviation is about 8. The regression then shows that we have a good fit between the two sets of flows.

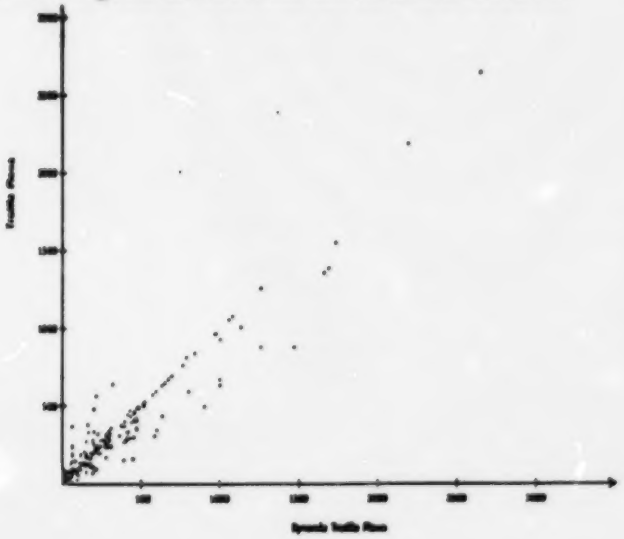

Figure 1

5. CONCLUSION

This paper is concerned with the Traffic Assignment Models (TAM) aimed at simulating the flows of urban transportation networks, mainly during the peak periods. All the proposed models of the literature fit into one of two main classes that are the Static (where the demand is assumed to be constant over the time), and the Dynamic Traffic Assignment Models (where this assumption is relaxed).

Altough the demand is not constant over the time in an urban context, the STAM's have been extensively used in practice; in fact, even if these models are not suitable to analyze the flow variations over the study horizon of time, they give a good estimation of the total flow on the links during the whole study horizon of time.

As for the dynamic models, they are more suitable to analyze the flow variations on the links during the peak periods, since the demand is allowed to vary in time. Unfortunately, most of the models of this class, proposed in the literature, incorporate important limitations; some do not apply to capacity constrained links, others allow only parallel paths, or only one destination, and finally some lead to huge problems, and then, can difficultly be used in practice.

This paper describes a model, in the class of the Dynamic Traffic Assignment Models, aimed at simulating the flows of urban transportation networks mainly during the peak periods, which is mainly based on the assumption that the users know the travel times on the links of the network, at each period (but not for the next periods), and that, at each node, they enter the actual shortest path; in other words, at the beginning of each period of time, a user is assigned on the first link of the actual shortest path to his destination. The numerical results that we obtained with the transportation network of the city of Hull (Quebec, Canada), show that our model is accurate, relatively fast (is mainly based on shortest path computations, and is fully parallelizable) and requires only a relatively little memory space for a computer implementation.

REFERENCES

Arnott, R., de Palma, A. and Lindsey, R. (1987) Departure Time and Route Choice for Routes in Parallel, Discussion Paper #658. Institute for Economic Research, Queen's University.

Carey, M. (1986) A Constraint Qualification for a Dynamic Traffic Assignment Model, *Transportation Science* 20, 55-58.

Carey, M. (1987) Optimal Time-Varing Flows on Congested Networks, *Operations Research* 35 No.1, 58-69.

Dafermos, S. and Sparrow, F.T. (1969) The Traffic Assignment Problem for a General Network, *Journal of Research of the National Bureau of Standards* 73B, 91-118.

de Palma, A., Hansen, P. and Labbé, M. (1987) Commuters' Paths with Penalities for Early or Late Arrival Time, Discussion Paper #8712. Centre for Operations Research & Econometrics, Université Catholique de Louvain.

Drissi-Kaïtouni, O. and Hameda-Benchekroun, A. (1989) A Dynamic Traffic Assignment Model and a Solution Algorithm, publication #649, Centre de recherche sur les transports, Université de Montréal.

Florian, M., Crainic, T.G., Guélat, J. and Babin, A. (1987) STAN : An Interactive-Graphic System for Strategic Transportation Analysis at the National and Regional Levels, publication #495, Centre de recherche sur les transports, Université de Montréal.

Ford, L.R. and Fulkerson, D.R. (1962) *Flows in Networks*, Princeton University Press. Princeton, New Jersey.

Friesz, T.L., Luque, J., Tobin, R.L. and Wie, B. (1988) Dynamic Network Traffic Assignment considered as a Continuous Time Optimal Control Problem, Department of Operations Research and Regional Science, University of Pennsylvania, Philadelphia, PA 19104.

Fulkerson, D.R. (1975) Studies in Graph Theory,part I, *Studies in Mathematics* 11, The Mathematical Association of America.

Guélat, J., Florian, M. and Crainic, T.G. (1987) A Multimode Multiproduct Network Assignment Model for Strategic Planning of Freight Flow, publication #549, Centre de recherche sur les transports, Université de Montréal.

Hendrickson, C. and Kocur, G. (1981) Schedule Delay and Departure Time Decision in a Determinisic Model, *Transportation Science* 15, 62-77

Ho, J.K. (1980) A Successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem, *Transportation Science* 14 No.4, 295-305.

Leonard, D.R., Tough, J.B. and Baguley, P.C. (1978) CONTRAM : A Traffic Assignment Model for Predecting Flows and Queues during Peak Periods, Transport and Road Research Laboratory, Report 841, Crowthorne.

Merchant, D.K. and Nemhauser, G.L. (1978) A Model and an Algorithm for the Dynamic Traffic Assignment Problem, *Transportation Science* 12 No.3, 183-199.

Merchant, D.K. and Nemhauser, G.L. (1978) Optimality Conditions for a Dynamic Traffic Assignment Model, *Transportation Science* 12 No.3, 200-207.

Robillard, P. (1974) Multipath Assignment with Dynamic Input Flows. *Transportation Research* 8, 567-573

Wardrop, J.G. (1952) Some Theoretical Aspects of Road Traffic Research, *Proceedings of the Institut of Civil Engineers*, Part II 1, 325-378.

Yagar, S. (1970) Dynamic Traffic Assignment by Individual Path Minimization and Queuing, *Transportation Research* 5, 179-196.

Zawack, D.J. and Thompson, G.L. (1987) A Dynamic Space-Time Network Flow Model for City Traffic Congestion, *Transportation Science* 21 No.3, 153-162.

APPENDIX 1

The link travel time functions used for the city of Hull (Quebec, Canada) are of the form:

$$c_a(x_a) \;=\; 1 \;+\; A_a\left(\frac{x_a}{Cap_a}\right)^{B_a} \quad \text{(expressed in periods of T0 minutes)}$$

where, for each link a,

$c_a(x_a)$	:	**is the travel time when the flow is x_a on link a.**
T0	:	**the length of a period of time (about 0.3 minute).**
Cap_a	:	**is the capacity of the link.**
A_a	:	**a factor (about 0.7).**
B_a	:	**a factor (about 4 to 5).**

In the following, we describe a method to construct the exit functions $g_a()$ (if they have not been calibrated *in situ*), from the above travel time functions $c_a()$.

The term $A_a\left(\frac{x_a}{Cap_a}\right)^{B_a}$, in the functions $c_a()$, expresses the delay caused by the congestion; if there is no congestion, the free traversal time being one period of time, the exit flow per unit of time is then all the flow on the link. In other words, $g_a(x_a) = x_a$ when $c_a(x_a)$ is about 1 period of time; this is illustrated in figure A1.1.

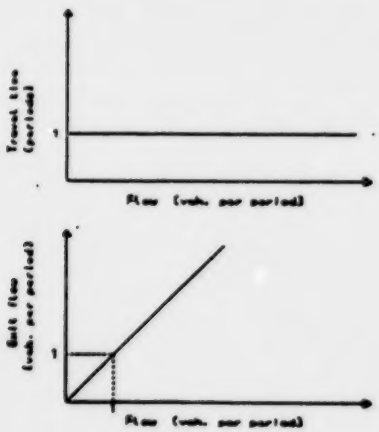

Figure A1.1: Exit function for an uncongested link.

While the congestion delay term $A_a\left(\frac{x_a}{Cap_a}\right)^{B_a}$ is close to 0 (say x_a is smaller than some $\bar{x}_a$), it is reasonable to keep using the exit function $g_a(x_a) = x_a$). But, for large values of the flow x_a (say x_a larger than some value $\bar{x}_a$ of the flow), an excess

delay of one period of time is due to an excess flow value of δx_a at $x_a - \delta x_a$ (see figure A1.2). In other words, when the value of the flow is x_a, the travel time is $c_a(x_a)$, and when the flow is $x_a - \delta x_a$, the travel time is $c_a(x_a) - 1$. Then, when the flow is x_a, the exit flow in one period of time ($c_a(x_a) - (c_a(x_a) - 1) = 1$), is δx_a ($x_a - (x_a - \delta x_a) = \delta x_a$); $g_a(x_a) = \delta x_a$ (see figure A1.2), where δx_a is computed easily by: $\delta x_a = x_a - c_a^{-1}(c_a(x_a) - 1)$.

For $\underline{x}_a \leq x_a \leq \bar{x}_a$, say $x_a = \alpha \underline{x}_a + (1-\alpha)\bar{x}_a$, we take $g_a(x_a) = \alpha g_a(\underline{x}_a) + (1-\alpha)g_a(\bar{x}_a)$. As for the values of $\underline{x}_a$ and $\bar{x}_a$, we used in our numerical tests $\underline{x}_a$ such that $c_a(\underline{x}_a) = 1.5$, and $\bar{x}_a$ such that $c_a(\bar{x}_a) = 2.5$. So, we used the functions:

$$g_a(x_a) = \begin{cases} x_a - c_a^{-1}(c_a(x_a) - 1) & \text{if } c_a(x_a) \geq c_a(\bar{x}_a) = 2.5 \\ x_a - (1-\alpha)c_a^{-1}(c_a(\bar{x}_a) - 1) & \text{if } x_a = \alpha \underline{x}_a + (1-\alpha)\bar{x}_a, \alpha \in [0,1] \\ x_a & \text{if } c_a(x_a) \leq c_a(\underline{x}_a) = 1.5 \end{cases}$$

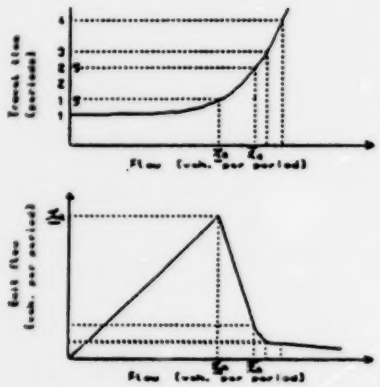

Figure A1.2: Relation between the delay and exit functions.

Example:

Let's consider the travel time function $c_a(x_a) = 1 + 0.7(\frac{x_a}{1000})^5$. Figure A1.3 illustrates $c_a()$ and the corresponding exit function $g_a()$. Assume that the initial flow is $x_a = 1338$ with a corresponding travel time $c_a(x_a)$ of 4 periods. According to the relations above, and assuming that no other flow enters the link, the exit flows will then be:

Period	x_a	$c_a(x_a)$	$g_a(x_a)$
1	1338	4	104
2	1234	3	160
3	1074	2	509
4	565	1.04	565
5	0	1	0
6	0	1	0
.	.	.	.

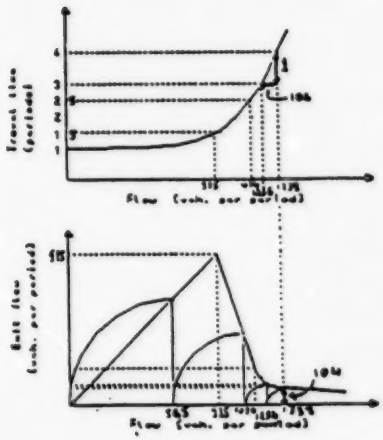

Figure A1.3: Example of an exit function.

APPENDIX 2

The total demand matrix for the city of Hull (Quebec, Canada) is given as follows: for each origin, there is a card of the form:

(Origin) (Nb. of Destinations) ((Destination(I),Demand(I)),I=1,Nb. of Destinations)

1	11	8	122	10	180	11	150	14	42	15	28	16	22	18	16
		19	16	20	9	21	34	23	13						
2	10	5	148	7	159	8	48	10	44	14	155	17	18	20	8
		21	38	22	21	23	11								
3	14	5	95	8	45	10	41	11	74	14	49	15	28	16	24
		17	23	18	16	19	16	20	5	21	27	22	58	23	8
4	9	5	257	7	349	14	116	18	12	19	42	20	24	21	100
		22	15	23	51										
5	13	2	782	3	40	4	1106	7	269	10	86	14	160	15	19
		16	10	17	52	18	16	19	36	20	22	22	60		
6	0														
7	10	2	164	4	1576	14	118	16	57	17	54	18	68	19	48
		20	23	21	54	23	34								
8	13	1	61	2	20	3	70	14	35	15	63	16	48	17	45
		18	60	19	43	20	8	21	176	22	21	23	43		
9	0														
10	13	1	352	2	30	3	106	14	55	15	66	16	49	17	18
		18	58	19	5	20	8	21	5	22	5	23	5		
11	10	1	590	3	111	14	110	15	100	16	32	17	21	18	27
		20	9	22	35	23	23								
12	0														
13	0														
14	0														
15	4	1	46	3	19	10	41	11	67						
16	4	1	46	3	19	10	31	11	41						
17	6	2	23	3	90	5	43	7	68	10	39	11	32		
18	7	1	54	2	20	3	32	5	64	7	29	10	45	11	40
19	5	1	54	2	10	3	42	8	60	10	37				
20	9	1	55	2	15	3	33	4	118	5	66	7	60	8	100
		10	29	11	40										
21	0														
22	4	2	15	3	10	4	17	11	10						
23	0														